UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/811,995	03/19/2001	Matthew J. Adiletta	10559-320001/P9681	9585

20985 7590 09/05/2007

FISH & RICHARDSON, PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
09/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 09/811,995	Applicant(s) ADILETTA ET AL.
	Examiner Aimee J. Li	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 June 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *17-21,23-26,28 and 29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *17-21,23-26,28 and 29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *7/18/2007*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 17-21, 23-26, and 28-29 were considered.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 05 June 2007 and IDS as filed 18 July 2007.

Information Disclosure Statement

3. The information disclosure statement (IDS) submitted on 18 July 2007 was filed after the mailing date of the Non-Final Rejection on 10 April 2007. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

5. A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

6. Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

7. Claims 17, 20-21, 23-24, 26, and 28-29 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 3, 9, and 10 of U.S. Patent No. 6,668,317 (herein referred to as '317) in view of Karguth, U.S. Patent Number 6,223,277 (herein referred to as Karguth) in further view of David K. Probst's "Programming, Compiling and Executing Partially-Ordered Instruction Streams on Scalable Shared-Memory Multiprocessors" from Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, 1994 ©1994 IEEE (herein referred to as Probst) and in further view of Tran, U.S. Patent Number 6,304,956 (herein referred to as Tran). Table I below shows the relations between the claims of the instant application, '317, Karguth, Probst, and Tran.

8. In regards to Karguth, a person of ordinary skill in the art at the time the invention was made, and as taught by Karguth, would have recognized that the packed data system maximizes on-chip utilization of memory and obtains performance at minimum cost (Karguth column 2, lines 50-65 "...As a result, packed data structures are attractive in these type of systems..."). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the packed data device of Karguth in the device of '317 to maximize on-chip memory utilization and minimize cost.

9. In regards to Probst, a person of ordinary skill in the art at the time the invention was made, and as taught by Probst, would have recognized that multithreading improves tolerance of latencies and increases processor utilization (Probst Section 2, paragraph 1 "Multithreading is commonly suggested as a technique for tolerating latencies and increasing processor utilization..."). Therefore, it would have been obvious to a person of ordinary skill in the art at

the time the invention was made to incorporate the multithreading of Probst in the device of '317 to improve latency tolerance and increase processor utilization.

10. In regards to Tran, a person of ordinary skill in the art at the time the invention was made would have recognized that the shift instructions of Tran, which include an operand designating the number of bits being shifted, allow more flexibility and customization from the user. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the shift instructions of Tran in the device of '317 for greater flexibility and customization from the user.

Table I

Instant Application	**Patent Number 6,668,317**
Claim 17	**Claims 1, 3, 9, 10**
A hardware-based multithreaded processor comprising:	*Probst* Section 2, paragraph 1 "Multithreading is commonly suggested as a technique…"
A plurality of microengines,	*Karguth* column 5, lines 12-16 "…another instance of network hub and ATM translator **5** would be implemented in place of ATM premises switch **8**…"; column 3, lines 40-43 "The present invention may be implemented in a micro-processor architecture…"; and Figure 1, element 5
Each of the microengines comprising	A microcontrolled functional execution unit comprises (Claim 1):
A context event arbiter,	A context event arbiter, which in response to external flags, determines which one of a plurality of threads executable in the microcontrolled functional execution unit to promote to an execution state (Claim 1).
A controller,	A microengine controller for maintaining a plurality of microprogram counters, and decode logic for decoding instructions (Claim 1);
A control store,	A control store to store a microprogram(Claim 1);
Local read and write transfer registers,	A read transfer register bank (Claim 9); and A write transfer register bank, with the read and write transfer register banks divided into a plurality of windows that correspond to the number of

	microprogram counters supported in the microengine controller (Claim 9) A read transfer register bank (Claim 10); and A write transfer register bank, with the read and write transfer register banks divided into a plurality of banks assigned for different shared resources in the microengine controller (Claim 10).
Local general purpose registers, and an arithmetic logic unit (ALU),	An arithmetic logic unit and shifter controlled by decoded signals produced from the microengine controller (Claim 3); and A general purpose register bank to store and obtain operands for the arithmetic logic unit (Claim 3).
Each of the microengines supporting instructions that	*Karguth* column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto…"; column 9, lines 27-32 "…provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed…"; column 9, lines 58 to column 10, line 9 "…contain an immediate operand value for use…contain a second source register…"; and Figure 3, element 30
perform an ALU operation on one or two operands,	*Karguth* column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto…"; column 9, lines 27-32 "…provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed…"; column 9, lines 58 to column 10, line 9 "…contain an immediate operand value for use…contain a second source register…"; and Figure 3, element 30
Deposit a result in a destination register and	Karguth column 8, lines 1-23 "…data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers…"; column 9, lines 27-32 "…provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed…"; and Figure 3

Update ALU condition codes according to the result; and	*Karguth* column 11, lines 35-51 "...the test is evaluated, thus resulting in a singlecycle test and branch instruction..." – In regards to Karguth, the test results essentially update the condition codes, eventhough they are not stored in a register, in order to determine if the condition is true or not and to branch accordingly.
A local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand,	*Karguth* column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; column 9, lines 58 to column 10, line 9 "...contain an immediate operand value for use...contain a second source register..."; and Figure 3, element 30
The field representing a mask in which each bit of the mask identifies a different byte of the destination register.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3
The shifted value of the other operand being shifted according to a shift control parameter specified by the local register instruction.	*Tran* column 1, line 11 to column 2, line 20 "...According to the Intel (TM) x86 instruction set, the shift instruction shifts the bits of the first operand (i.e. destination operand) to the left or right by the number of bits specified in the second operand (i.e. count operand)..."
Claim 20	
The processor of claim 17, wherein the destination register is a general purpose register.	*Karguth* column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3

Claim 21	
The processor of claim 17, wherein the local register instruction comprises the destination register.	*Karguth* column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3
Claim 23	
The processor of claim 17, wherein the mask is 4-bits.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3 – In regards to Karguth, the exemplary mask is 3-bits, however, the size of the mask field does not matter and it is only an exemplary embodiment.
Claim 24	
The processor of claim 17, wherein the mask comprises a set bit indicating a corresponding byte in the local register to be loaded.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3
Claim 26	
Apparatus comprising:	
A hardware-based multithreaded processor comprising	*Probst* Section 2, paragraph 1 "Multithreading is commonly suggested as a technique..."
A plurality of microengines,	*Karguth* column 5, lines 12-16 "...another instance

	of network hub and ATM translator **5** would be implemented in place of ATM premises switch **8**…"; column 3, lines 40-43 "The present invention may be implemented in a micro-processor architecture…"; and Figure 1, element 5
Each of the microengines comprising	A microcontrolled functional execution unit comprises (Claim 1):
A context event arbiter,	A context event arbiter, which in response to external flags, determines which one of a plurality of threads executable in the microcontrolled functional execution unit to promote to an execution state (Claim 1).
A controller,	A microengine controller for maintaining a plurality of microprogram counters, and decode logic for decoding instructions (Claim 1);
A control store,	A control store to store a microprogram(Claim 1);
Local read and write transfer registers,	A read transfer register bank (Claim 9); and A write transfer register bank, with the read and write transfer register banks divided into a plurality of windows that correspond to the number of microprogram counters supported in the microengine controller (Claim 9) A read transfer register bank (Claim 10); and A write transfer register bank, with the read and write transfer register banks divided into a plurality of banks assigned for different shared resources in the microengine controller (Claim 10).
Local general purpose registers, and an arithmetic logic unit (ALU),	An arithmetic logic unit and shifter controlled by decoded signals produced from the microengine controller (Claim 3); and A general purpose register bank to store and obtain operands for the arithmetic logic unit (Claim 3).
Each of the plurality of microengines including a command that causes the ALU to load one or more bytes, specified by a multiple-bit field of the command, within a destination register of a selected microengine with a shifted value of another one or more bytes of a source register,	*Karguth* column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto…"; column 9, lines 27-32 "…provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed…"; column 9, lines 58 to column 10, line 9 "…contain an immediate operand value for use…contain a second source register…"; and Figure

	3, element 30 *Karguth* column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; column 9, lines 58 to column 10, line 9 "...contain an immediate operand value for use...contain a second source register..."; and Figure 3, element 30 *Karguth* column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3 *Karguth* column 11, lines 35-51 "...the test is evaluated, thus resulting in a singlecycle test and branch instruction..." – In regards to Karguth, the test results essentially update the condition codes, eventhough they are not stored in a register, in order to determine if the condition is true or not and to branch accordingly. *Karguth* column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; column 9, lines 58 to column 10, line 9 "...contain an immediate operand value for use...contain a second source register..."; and Figure 3, element 30
The field representing a mask in which each bit of the mask identifies a different byte of the destination register.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units

	28…"; column 10, lines 10-48 "…write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word…so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)…" and Figure 3
The shifted value of the other operand being shifted according to a shift control parameter specified by the local register instruction.	*Tran* column 1, line 11 to column 2, line 20 "…According to the Intel (TM) x86 instruction set, the shift instruction shifts the bits of the first operand (i.e. destination operand) to the left or right by the number of bits specified in the second operand (i.e. count operand)…"
Claim 28	
The apparatus of claim 26, wherein the mask is 4-bits.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**…"; column 10, lines 10-48 "…write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word…so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)…" and Figure 3 – In regards to Karguth, the exemplary mask is 3-bits, however, the size of the mask field does not matter and it is only an exemplary embodiment.
Claim 29	
The apparatus of claim 26, wherein the mask comprises a set bit indicating a corresponding byte in the source register to be loaded.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**…"; column 10, lines 10-48 "…write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word…so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)…" and Figure 3

11. Claims 17-21, 23-26, and 28-29 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 2-3, and 5-6 of U.S. Patent No. 7,191,309 (herein referred to as '309) in view of Karguth, U.S. Patent Number 6,223,277 (herein referred to as Karguth) and in further view of Tran, U.S. Patent Number 6,304,956 (herein referred to as Tran). Table II below shows the relations between the claims of the instant application, '309, Karguth, and Tran.

12. In regards to Karguth, a person of ordinary skill in the art at the time the invention was made, and as taught by Karguth, would have recognized that the packed data system maximizes on-chip utilization of memory and obtains performance at minimum cost (Karguth column 2, lines 50-65 "...As a result, packed data structures are attractive in these type of systems...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the packed data device of Karguth in the device of '309 to maximize on-chip memory utilization and minimize cost.

13. In regards to Tran, a person of ordinary skill in the art at the time the invention was made would have recognized that the shift instructions of Tran which include an operand designating the number of bits being shifted allow more flexibility and customization from the user.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the shift instructions of Tran in the device of '317 for greater flexibility and customization from the user.

Table II

Instant Application	**Patent Number 7,191,309**
Claim 17	**Claim 1**
A hardware-based multithreaded processor comprising:	A hardware-based multithreaded processor comprising:

A plurality of microengines,	A plurality of microengines,
Each of the microengines comprising	Each of the microengines comprising:
A context event arbiter,	Context event switching logic;
A controller,	Controller logic;
A control store,	A control store
Local read and write transfer registers,	*Karguth* column 8, lines 12-17 "...memory interface **37**, for writing results of the operation to parameter memory **18** over buses MEMD, MEMA..." and Figure 3, element 37 – In regards to Karguth, the memory interface must hold, e.g. storing them in registers, the data in some way in order to transfer data to and from parameter memory without it incorrectly changing and/or influencing the rest of the system.
Local general purpose registers, and an arithmetic logic unit (ALU),	An execution box data path including an arithmetic logic unit (ALU) and a general purpose register set,
Each of the microengines supporting instructions that	The ALU performing functions in response to instructions
perform an ALU operation on one or two operands,	One of the instructions causing the ALU to load a destination register with a 32-bit word formed by concatenating a first operand and a second operand to form a 64-bit result
Deposit a result in a destination register and	One of the instructions causing the ALU to load a destination register with a 32-bit word formed by concatenating a first operand and a second operand to form a 64-bit result
Update ALU condition codes according to the result; and	*Karguth* column 11, lines 35-51 "...the test is evaluated, thus resulting in a singlecycle test and branch instruction..." – In regards to Karguth, the test results essentially update the condition codes, eventhough they are not stored in a register, in order to determine if the condition is true or not and to branch accordingly.
A local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand,	One of the instructions causing the ALU to load a destination register with a 32-bit word formed by concatenating a first operand and a second operand to form a 64-bit result, shifting the 64-bit result by a specified amount, and storing a lower 32-bits of the 64-bit result.
The field representing a mask in which each bit of the mask identifies a different byte of the destination register.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of

	the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3
The shifted value of the other operand being shifted according to a shift control parameter specified by the local register instruction.	*Tran* column 1, line 11 to column 2, line 20 "...According to the Intel (TM) x86 instruction set, the shift instruction shifts the bits of the first operand (i.e. destination operand) to the left or right by the number of bits specified in the second operand (i.e. count operand)..."
Claim 18	**Claim 5**
The processor of claim 17, wherein the destination register is an absolute transfer register.	The processor of claim 1 wherein the destination register is an absolute register name.
Claim 19	**Claim 6**
The processor of claim 17, wherein the destination register is a context-relative transfer register.	The processor of claim 1 wherein the destination register is a context relative register name.
Claim 20	
The processor of claim 17, wherein the destination register is a general purpose register.	*Karguth* column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3
Claim 21	
The processor of claim 17, wherein the local register instruction comprises the destination register.	*Karguth* column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3
Claim 23	
The processor of claim 17, wherein the mask is 4-bits.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back

	operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3 – In regards to Karguth, the exemplary mask is 3-bits, however, the size of the mask field does not matter and it is only an exemplary embodiment.
Claim 24	
The processor of claim 17, wherein the mask comprises a set bit indicating a corresponding byte in the local register to be loaded.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3
Claim 25	**Claims 2 and 3**
The processor of claim 17, wherein the local register comprises a context relative source register.	The processor of claim 1 wherein the first operand is a context-relative 32-bit register. The processor of claim 1 wherein the second operand is a context-relative 32-bit register.
Claim 26	**Claim 1**
Apparatus comprising:	
A hardware-based multithreaded processor comprising	A hardware-based multithreaded processor comprising:
A plurality of microengines,	A plurality of microengines,
Each of the microengines comprising	Each of the microengines comprising:
A context event arbiter,	Context event switching logic; and
A controller,	Controller logic;
A control store,	A control store;
Local read and write transfer registers,	*Karguth* column 8, lines 12-17 "...memory interface **37**, for writing results of the operation to parameter memory **18** over buses MEMD, MEMA..." and Figure 3, element 37 – In regards to Karguth, the memory interface must hold, e.g. storing them in registers, the data in some way in order to transfer data to and from

	parameter memory without it incorrectly changing and/or influencing the rest of the system.
Local general purpose registers, and an arithmetic logic unit (ALU),	An execution box data path including an arithmetic logic unit (ALU) and a general purpose register set,
Each of the plurality of microengines including a command that causes the ALU to load one or more bytes, specified by a multiple-bit field of the command, within a destination register of a selected microengine with a shifted value of another one or more bytes of a source register,	The ALU performing functions in response to instructions, one of the instructions causing the ALU to load a destination register with a 32-bit word formed by concatenating a first operand and a second operand to form a 64-bit result, shifting the 64-bit result by a specified amount, and storing a lower 32-bits of the 64-bit result.
The field representing a mask in which each bit of the mask identifies a different byte of the destination register.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**…"; column 10, lines 10-48 "…write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word…so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)…" and Figure 3
The shifted value of the other operand being shifted according to a shift control parameter specified by the local register instruction.	*Tran* column 1, line 11 to column 2, line 20 "…According to the Intel (TM) x86 instruction set, the shift instruction shifts the bits of the first operand (i.e. destination operand) to the left or right by the number of bits specified in the second operand (i.e. count operand)…"
Claim 28	
The apparatus of claim 26, wherein the mask is 4-bits.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**…"; column 10, lines 10-48 "…write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word…so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)…" and Figure 3 – In regards to Karguth, the exemplary mask is 3-bits, however, the size of the mask field

	does not matter and it is only an exemplary embodiment.
Claim 29	
The apparatus of claim 26, wherein the mask comprises a set bit indicating a corresponding byte in the source register to be loaded.	*Karguth* column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 17, 20-21, 23-26, and 28-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Karguth, U.S. Patent Number 6,223,277 (herein referred to as Karguth) in view of David K. Probst's "Programming, Compiling and Executing Partially-Ordered Instruction Streams on Scalable Shared-Memory Multiprocessors" from Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, 1994 ©1994 IEEE (herein referred to as Probst) and in further view of Tran, U.S. Patent Number 6,304,956 (herein referred to as Tran).

16. Referring to claims 17 and 26, taking claim 17 as exemplary, Karguth has taught a processor comprising:

a. A plurality of microengines (Karguth column 5, lines 12-16 "…another instance of network hub and ATM translator **5** would be implemented in place of ATM premises switch **8**…"; column 3, lines 40-43 "The present invention may be implemented in a micro-processor architecture…"; and Figure 1, element 5),

b. Each of the microengines comprising

   i. A controller (Karguth column 7, lines 47-66 "…decoded by control and instruction decode circuitry **32**…" and Figure 3, element 32),

   ii. A control store (Karguth column 7, lines 47-66 "…under the control of instructions retrieved from instruction memory **38**…" and Figure 3, element 32),

   iii. Local read and write transfer registers (Karguth column 8, lines 12-17 "…memory interface **37**, for writing results of the operation to parameter memory **18** over buses MEMD, MEMA…" and Figure 3, element 37 – In regards to Karguth, the memory interface must hold, e.g. storing them in registers, the data in some way in order to transfer data to and from parameter memory without it incorrectly changing and/or influencing the rest of the system.),

   iv. Local general purpose registers (Karguth column 7, lines 12-29 "…Each of the remaining registers REG**0** through REG**30** are general purpose registers…" and Figure 3, element 24), and

c. An arithmetic logic unit (ALU) (Karguth column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto..." and Figure 3, element 30),

d. Each of the microengines supporting instructions that perform an ALU operation on one or two operands (Karguth column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; column 9, lines 58 to column 10, line 9 "...contain an immediate operand value for use...contain a second source register..."; and Figure 3, element 30), deposit a result in a destination register (Karguth column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3) and update ALU condition codes according to the result (Karguth column 11, lines 35-51 "...the test is evaluated, thus resulting in a singlecycle test and branch instruction..." – In regards to Karguth, the test results essentially update the condition codes, eventhough they are not stored in a register, in order to determine if the condition is true or not and to branch accordingly.); and

e. A local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand (Karguth column 7, lines 47-67 "ALU **30** includes the appropriate circuitry for executing arithmetic and logical operations upon operands presented thereto..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; column 9, lines 58 to column 10, line 9 "...contain an immediate operand value for use...contain a second source register..."; and Figure 3, element 30),

f. The field representing a mask in which each bit of the mask identifies a different byte of the destination register (Karguth column 9, lines 33-38 "Bit position **7**:**5** and **15**:**13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3).

17. Karguth has not taught a hardware-based multithreaded and a context event arbiter. Probst has taught a hardware-based multithreaded (Probst Section 2, paragraph 1 "Multithreading is commonly suggested as a technique...") and a context event arbiter (Probst Section 2, paragraph 4 "...Multithreaded architectures differ in their context switching

policies..." and Section 2, paragraphs 5-6 "Block multithreaded processors, which switch contexts only when a high-latency operation is encountered..." – In regards to Probst, the mechanism controlling the context switching as taught by Probst is a context event arbiter.). A person of ordinary skill in the art at the time the invention was made, and as taught by Probst, would have recognized that multithreading improves tolerance of latencies and increases processor utilization (Probst Section 2, paragraph 1 "Multithreading is commonly suggested as a technique for tolerating latencies and increasing processor utilization..."). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the multithreading of Probst in the device of Karguth to improve latency tolerance and increase processor utilization.

18. In addition, Karguth has not explicitly taught the shifted value of the other operand being shifted according to a shift control parameter specified by the local register instruction. However, Karguth has taught in the cited section above shift instructions in general, shift instruction execution, and that instructions in general contain an immediate operand, but not the details of the shift instructions nor what the immediate operand in an instruction represents. Tran has explicitly taught the shifted value of the other operand being shifted according to a shift control parameter specified by the local register instruction (Tran column 1, line 11 to column 2, line 20 "...According to the Intel (TM) x86 instruction set, the shift instruction shifts the bits of the first operand (i.e. destination operand) to the left or right by the number of bits specified in the second operand (i.e. count operand)..."). In regards to Tran, the count operand is an immediate value in the shift instruction. A person of ordinary skill in the art at the time the invention was made would have recognized that the shift instructions of Tran which include an

operand designating the number of bits being shifted allow more flexibility and customization from the user. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the shift instructions of Tran in the device of '317 for greater flexibility and customization from the user.

19. Claim 26 has similar limitations to claim 17 and is rejected for similar reasons. Claim 26 differs from claim 17 only in that claim 26 is for an apparatus while claim 17 is a processor.

20. Referring to claim 20, Karguth in view of Probst in further view of Tran has taught the processor of claim 17, wherein the destination register is a general purpose register (Karguth column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3).

21. Referring to claim 21, Karguth in view of Probst in further view of Tran has taught the processor of claim 17, wherein the local register instruction comprises the destination register (Karguth column 8, lines 1-23 "...data results that are to be written back into register file **24** are applied to shifter **34**, to place the operand in the appropriate bit positions of the destination of one of registers..."; column 9, lines 27-32 "...provide a five-bit selection code by way of which the destination and source registers, respectively, for the instruction are addressed..."; and Figure 3).

22. Referring to claims 23 and 28, taking claim 23 as exemplary, Karguth in view of Probst in further view of Tran has taught the processor of claim 17, wherein the mask is 4-bits (Karguth

column 9, lines 33-38 "Bit position **7:5** and **15:13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3 – In regards to Karguth, the exemplary mask is 3-bits, however, the size of the mask field does not matter and it is only an exemplary embodiment.).

23. Claim 28 has similar limitations to claim 23 and is rejected for similar reasons. Claim 28 differs from claim 23 only in that claim 28 is for an apparatus while claim 23 is a processor.

24. Referring to claims 24 and 29, taking claim 24 as exemplary, Karguth in view of Probst in further view of Tran has taught the processor of claim 17, wherein the mask comprises a set bit indicating a corresponding byte in the local register to be loaded (Karguth column 9, lines 33-38 "Bit position **7:5** and **15:13** each provide a three-bit code, by way of which the desired portion of the destination and source registers are to be selected by the shift/mask units **28**..."; column 10, lines 10-48 "...write-back operation according to the preferred embodiment of the present invention also shifts the output result from ALU **30** into the proper position within the result word...so that one or more of the byte locations may be written with the contents of writeback bus WBBUS (the remaining bits being masked from the write)..." and Figure 3).

25. Claim 29 has similar limitations to claim 24 and is rejected for similar reasons. Claim 29 differs from claim 24 only in that claim 29 is for an apparatus while claim 24 is a processor.

26.

27. Claims 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Karguth, U.S. Patent Number 6,223,277 (herein referred to as Karguth) in view of David K. Probst's "Programming, Compiling and Executing Partially-Ordered Instruction Streams on Scalable Shared-Memory Multiprocessors" from Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, 1994 ©1994 IEEE (herein referred to as Probst) in further view of Tran, U.S. Patent Number 6,304,956 (herein referred to as Tran), as applied to claim 17 above, and further in view of Vincent P. Heuring and Harry F. Jordan's Computer Systems Design and Architecture ©1997 (herein referred to as Heuring).

28. Referring to claims 18, 19, and 25, Karguth in view of Probst in further view of Tran has not explicitly taught

a. The processor of claim 17, wherein the destination register is an absolute transfer register (Applicant's claim 18).

b. The processor of claim 17, wherein the destination register is a context-relative transfer register (Applicant's claim 19).

c. The processor of claim 17, wherein the local register comprises a context-relative source register (Applicant's claim 25).

29. However, Karguth has taught accessing registers but not specifically how these registers are accessed, i.e. that the registers are addressed via absolute transfers or context-relative transfers. Heuring has taught

a. The processor of claim 17, wherein the destination register is an absolute transfer register (Applicant's claim 18) (Heuring pages 69-71, Table 2.8).

b. The processor of claim 17, wherein the destination register is a context-relative transfer register (Applicant's claim 19) (Heuring pages 69-71, Table 2.8).

c. The processor of claim 17, wherein the local register comprises a context relative source register (Applicant's claim 25) (Heuring pages 69-71, Table 2.8).

30. A person of ordinary skill in the art at the time the invention was made would have recognized that have an addressing method for the registers ensures that the correct registers are accessed when data needs to be retrieved, i.e. read, or written, i.e. stored, in the registers. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the register address schemes of Heuring in the device of Karguth to ensure the correct registers are accessed.

Response to Arguments

31. Applicant's arguments with respect to claims 17-21, 23-26, and 28-29 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

32. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

a. Sidwell, U.S. Patent Numbers 5,822,619; 5,859,789; 5,859,790; 5,875,355; 5,884,069; 6,100,905; and 6,145,077 have taught packed data instructions,

including shift instructions that have an immediate operand designating how many locations are to be shifted, and execution of these instructions.

33. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

34. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

35. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

36. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

37. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Aimee J Li
Examiner
Art Unit 2183

30 August 2007

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100